TECHNIQUE FOR OVERSAMPLING TO REDUCE JITTER

CROSS REFERENCE TO RELATED APPLICATION

This application claims priority under 35 U.S.C. 119(e) to U.S. Provisional Patent Application Serial No 60/454,582, filed March 21, 2003, the teachings of which are incorporated herein.

TECHNICAL FIELD

This invention relates to a technique for sampling an asynchronous digital signal to achieve reduced jitter performance

BACKGROUND ART

In various types of electronic systems, a need exists to sample a digital signal to determine its state, i.e., to determine whether the signal is at a logical "1" or logical "0" level. In practice, such sampling occurs periodically, typically in response to a periodic sample clock pulse. Ideally, sampling should occur at a sufficiently high frequency to minimize *jitter*, generally defined as the period of uncertainty between a change in state of the sampled signal and the next sample clock pulse. In a worst-case scenario, the sampled signal will change states just after the occurrence of a sample clock, so that nearly an entire sample clock interval will elapse before sampling the input signal again.

Increasing the sampling frequency will reduce the period of uncertainty and thus yield improved jitter performance. However, increasing the sampling frequency will yield more samples. In some electronic systems, bandwidth constraints limit the number samples capable of being transmitted within a given interval. In such systems, limited opportunities exist for jitter performance improvement.

Thus, there is need for technique that achieves increased jitter performance in bandwidth-limited systems.

BRIEF SUMMARY OF THE INVENTION

Briefly, in accordance with a preferred embodiment of the present principles, there is provided a method for sampling a digital signal that yields improved jitter performance. The method commences by sampling the digital signal n times during a given interval, with n chosen such that $\log_2(n+1)$ is an integer x greater than zero. After each interval t, an $x+1$ bit data block is generated. The first bit represents the state of the sampled signal, and the

remaining x bits indicate the sample interval (i.e., the particular one of the n clock pulses) during which a change, if any, occurred. If a change did occur, then sample value is inverted upon decoding to coincide with the change in the sample value that occurred during the indicated sample interval.

BRIEF SUMMARY OF THE DRAWINGS

FIGURE 1 depicts a block schematic diagram of an apparatus in accordance with the prior art for sampling an asynchronous digital signal;

FIGURE 2 depicts a time line showing the relationship over time among the digital signal undergoing sampling, the sample value obtained and the sample clock pulses generated by a sample clock comprising part of the apparatus of FIG. 1;

FIGURE 3 depicts a block schematic diagram of an apparatus in accordance with the present principles for sampling an asynchronous digital signal with improved jitter performance; and

FIGURE 4 depicts a time line showing the relationship over time among the digital signal undergoing sampling, the sample value obtained and the sample clock pulses generated by a sample clock comprising part of the apparatus of FIG. 3.

DETAILED DESCRIPTION

FIGURE 1 depicts a prior art apparatus 10 for sampling an asynchronous digital signal 11 generated by a transmitter 12. The transmitter 12 can take the form of any type of digital device that produces an asynchronous digital output signal. In other words, the output signal 11 of the transmitter 12 changes states between a logic "1" level and a logic "0" level a periodically. The sampling apparatus 10 includes a receiver 14 coupled to the output of the transmitter 12 for detecting the state of the signal 11 in response each of a sequence of periodic clock pulses 15 from a sample clock 16. The sample clock 16 generates m uniformly spaced clock pulses 15 during an interval of duration t. In the illustrated embodiment, m=5 although the value of m could be larger or smaller.

FIGURE 2 depicts a timing chart that shows the relationship over time between the output signal 11 of the transmitter 12 of FIG. 1 and the clock pulses 15 generated by the sample clock 16 of FIG. 1. In the example depicted in FIG. 2, the signal 11 undergoes a transition from a logic "1" state to a logic "0" level shortly after the occurrence of the second of the clock pulses 15 generated by the sample clock 16 of FIG. 1. Following receipt of the

second clock pulse 15, the receiver 14 of FIG. 1 will again sample the signal 11. Upon finding the signal 11 at a logic "1" level, the receiver 14 will maintain its output signal 20 in FIG. 2 at a logic "1" level as before. The output signal 20 of the receiver 14 remains at this logic state until the next sampling interval following receipt of the next (i.e., the third) clock pulse 15. When clocked again by the sample clock 16 of FIG. 1 (i.e., the fourth clock pulse occurrence), the receiver 14 of FIG. 2 now finds that the amplitude of the signal 11 has dropped to a logic "0" level. Accordingly, the receiver output signal 20 shown in FIG. 2 now drops to a logic "0" level.

The shaded portion of the receiver output signal 20 depicted in FIG. 2 corresponds to the lag between the change in state of the signal 11 and the change in state of the receiver output signal. This lag, referred to in the art as *jitter*, results from the periodic sampling of an asynchronous signal and can last up to one clock cycle. Increasing the sampling rate (i.e., increasing the number of sample clock pulses per time interval t) will yield more samples, thus reducing the amount of jitter. However, some electronic systems have bandwidth constraints that limit the number of samples capable of being transmitted during a given interval. In such systems, the option of increasing the sampling rate to improve jitter performance does not exist.

FIGURE 3 depicts a block schematic diagram of a system 100, in accordance with a preferred embodiment of the present principles, for sampling an asynchronous digital signal 11 produced by transmitter 12 to achieve improved jitter performance while maintaining bandwidth constraints. The system 100 includes a receiver 140 that samples the output signal 11 upon receipt of each of a sequence of periodic clock pulses 150 generated by a clock 160. As compared to the sample clock 16 of FIG. 1 which generates m uniformly spaced periodic pulses 15 during each interval t, the sample clock 160 of FIG. 3 generates n uniformly spaced pulses 150 during the same interval t where $n > m$. In the illustrated embodiment, $n = 15$, although n could assume larger or smaller values, so long as $\log_2(n+1)$ is an integer x greater than zero. Rather than generating a single bit sample value like the receiver 14 of FIG. 1, the receiver 140 of FIG. 3 generates a sample comprised of an $x+1$ bit word 200 during each interval t. The $x+1$ bit word 200 indicates the sample value of the signal 11 and also identifies the particular one of the n clock cycles within the interval t during the signal 11 underwent a change in state if such a change did in fact occur.

FIGURE 4 depicts a timing chart that shows the relationship over time between the signal 11 generated by the transmitter 12 of FIG 2 and the pulses 150 generated by the clock

160. As shown in FIG. 4, during each sample interval of duration *t*, the receiver 140 of FIG. 1 receives a successive one of fifteen clock pulses 150, each pulse causing the receiver to sample the input signal 11. At the end of every interval t, the receiver 140 generates a five-bit word 200 whose first bit (hereinafter referred to as a "reference data" bit) indicates the state of the input signal 11. The remaining four bits collectively identity of the particular one of the fifteen clock cycles within the interval t during which the signal 11 changed state, assuming such a change occurred.

As can be appreciated from the timing chart depicted in FIG. 4, the last four bits of the five-bit sample will uniquely identify each of the fifteen clock pulses within the interval *t*. Thus for example, the four-bit binary word (0 0 0 0) serves to identify the first of the fifteen pulses when there is no change in the fifteen-clock pulse period, whereas the four bit-binary word (1 1 1 1) identifies the fifteenth clock pulse when a change occurred at that time. In this way, the five-bit sample generated by the receiver 140 of FIG. 3 provides a single sample during each interval *t* with the same precision as fifteen individual samples, thus, improving jitter performance without violating bandwidth constraints. If a change did occur in the value of the signal, then sample value in the five-bit word 200 is inverted upon decoding to coincide with the change in the sample that occurred.

The sampling method performed by the receiver 140 constitutes a form of run-length encoding which yields a single sample (referred to as a Run Length Encoded Data Block or RLEDB) of *x* bits in length with the same precision as *n* samples, provided that $\log_2(n+1) = x$. This scheme trades the maximum sample acquisition rate of a given communications link bandwidth for improved jitter performance. In the illustrated embodiment, a compression factor of 3.2-to-1 is achieved (16 samples in five periods), so jitter is reduced by the same ratio. Other compression factors are possible.

The only drawback of this method is that the signal 11 cannot undergo a change in state more than once during each interval *t*. In other words, the signal 11 cannot undergo more than one change for each RLEDB. If the signal 11 were to change states two or more times per RLEDB, data will become lost.

The foregoing describes a technique for sampling an asynchronous signal with improved jitter performance while maintaining bandwidth constraints.